PATENT
P56996

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

HYUN-SOOK LEE

Serial No.:	*to be assigned*	Examiner:	*to be assigned*
Filed:	13 February 2004	Art Unit:	*to be assigned*

For: SECURITY METHOD FOR OPERATOR ACCESS CONTROL OF NETWORK MANAGEMENT SYSTEM

CLAIM OF PRIORITY UNDER 35 U.S.C. §119

Mail Stop : Patent Application
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

The benefit of the filing date of the following prior foreign application, Korean Priority Nos.2003-10509 (filed in Korea on 19 February 2003) and 2003-34534 (filed in Korean On 29 May 2003), and filed in the U.S. Patent and Trademark Office on 13 February 2004 is hereby requested and the right of priority provided in 35 U.S.C. §119 is hereby claimed.

In support of this claim, filed herewith is certified copies of said original foreign applications.

Respectfully submitted,

Robert E. Bushnell
Reg. No.: 27,774
Attorney for the Applicant

1522 "K" Street, N.W., Suite 300
Washington, D.C. 20005
(202) 408-9040
Folio: P56996
Date: 2/13/04
I.D.: REB/rfc

대 한 민 국 특 허 청
KOREAN INTELLECTUAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출 원 번 호 : 10-2003-0010509
Application Number

출 원 년 월 일 : 2003년 02월 19일
Date of Application FEB 19, 2003

출 원 인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.

2003 년 09 월 26 일

특 허 청
COMMISSIONER

【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【제출일자】	2003.02.19
【발명의 명칭】	망관리 시스템의 운용자 접근 제어에 대한 보안 방법
【발명의 영문명칭】	Security method for access control of Network management System
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	박상수
【대리인코드】	9-1998-000642-5
【포괄위임등록번호】	2000-054081-9
【발명자】	
【성명의 국문표기】	이현숙
【성명의 영문표기】	LEE,HYUN SOOK
【주민등록번호】	700626-2110829
【우편번호】	449-753
【주소】	경기도 용인시 수지읍 동성1차아파트 108동 1305호
【국적】	KR
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사 를 청구합니다. 대리인 박상수 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	2 면 2,000 원
【우선권주장료】	0 건 0 원
【심사청구료】	4 항 237,000 원
【합계】	268,000 원
【첨부서류】	1. 요약서·명세서(도면)_1통

1020030010509 출력 일자: 2003/10/1

【요약서】

【요약】

본 발명은 현장에서 사용하고 있는 시스템 어플리케이션 프로토콜의 버전을 바꾸지 않고도 접근 제어(access control)를 할 수 있는 방법을 제공하기 위한 것으로서, 운용자는 사용자 인증을 위해 아이디(ID)와 패스워드(password)를 입력하고, 사용자 인증과정이 성공되면, TCP/IP 또는 UDP/IP를 통해 관리하고자 하는 시스템의 어플리케이션 계층에 접근하게 되는데, 이때 운용자가 사용하는 단말기의 IP 주소가 미리 설정된 IP 주소인지를 확인하기 위해, 보안 모듈을 통해 어플리케이션 계층에 접근하도록 구성됨으로써, 보안 기능이 없는 망관리 인터페이스 버전을 운용하는 망의 경우 버전 업그레이드 과정없이 간단하게 보안 모듈을 추가함으로써 시스템의 보안 문제를 해결해준다.

【대표도】

도 3

【색인어】

SNMP, CLI, TL1, 프로토콜 보안, OAMP

【명세서】

【발명의 명칭】

망관리 시스템의 운용자 접근 제어에 대한 보안 방법{Security method for access control of Network management System}

【도면의 간단한 설명】

도 1은 본 발명에 적용되는 간이 망 관리 프로토콜(SNMP) 및 CLI(TL1)을 이용한 망관리 시스템을 설명하기 위한 블록도,

도 2는 종래 기술에 따른 망관리 시스템을 OSI 참조 모델과 관련하여 설명한 도면,

도 3은 본 발명에 따른 망관리 시스템을 OSI 참조 모델과 관련하여 설명한 도면.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<4> 본 발명은 시스템 어플리케이션 프로토콜의 버전을 바꾸지 않고도 접근제어를 할 수 있도록 하는 망관리 시스템의 운용자 접근 제어에 대한 보안 방법에 관한 것이다.

<5> 현재, 인터넷을 포함한 네트워크 관련 장비들은 대부분 네트워크를 관리하고, 각 네트워크 장치들의 동작을 감시하기 위해 간이 망 관리 프로토콜 (Simple Network Management Protocol: SNMP)을 기반으로 하는 네트워크 관리 프로토콜을 이용한다. 이러한 SNMP는 가장 일반적인 네트워크 관리 프로토콜이라 할 수 있으며, 각각 V1/V2/V3의 버전으로 발전되면서 그 기능이 많이 향상되었다. 대부분의 네트워크 시스템은 이러한 SNMP를 이용하는 그래픽 운용자 인터페이스 (Graphic User Interface: GUI) 기반의 엘리먼트 관리 시스템(Element Management

system:EMS)과 외부 터미널을 통하여 명령어를 직접 수신하여 처리하는 명령어 인터페이스(Command Line Interface :CLI)를 함께 제공한다.

<6> 도 1은 본 발명에 적용되는 망 관리 프로토콜(SNMP) 및 CLI(TL1)을 이용한 망관리 시스템을 설명하기 위한 블록도이다. 도 1을 참조하면, 시스템(100)에서 제공하는 망 관리 인터페이스(management interface)에는 "TL1/CLI(Transaction Language 1/Command Line Interface)(110)"과 "SNMP 대리인(120)"이 있다. 시스템은 이러한 관리 채널(management channel)들을 통해 시스템의 구성, 경보, 성능 등을 관리하게 된다.

<7> TL1(110)의 경우, 외부의 콘솔(200)들과 시리얼 포트(serial port)로 직접 연결하여 시스템(100)을 관리할 수도 있고, 공중망(public network)(300)을 통한 텔넷(telnet)(400)으로 원격 제어(remote control)도 가능하다.

<8> 한편, SNMP 대리인(120)은 UDP(user datagram protocol)/IP를 이용하여 공중망(300)을 통해 주로 EMS(Element Management system) 서버(500)와 연결되어 사용되어지나, OSI(open systems interconnection) CLNP(Connectionless Network Protocol) 위의 프로토콜로 사용되어지기도 한다.

<9> 이와 같은 TL1(110)과 SNMP 대리인(120)은 각각 OAMP(Operations Administration Maintenance Provisioning)(130)로부터, 또는 IPC(Interprocess Communication)를 통해 필요한 데이터를 가지고 오기도 하고, 변경시키기도 한다.

<10> 도 2는 종래 기술에 따른 망관리 시스템을 OSI 참조 모델과 관련하여 설명한 도면이다. 도 2를 참조하면, 텔넷 터미널(400) 또는 EMS 서버(500)는 물리 계층을 통해 데이터링크 계층

과 연결되어, TCP/IP 또는 UDP/IP 방식으로 어플리케이션 계층{SNMP/ 텔넷/ TFTP(Trivial File Transfer Protocol)}에 접근하게 된다.

<11> 상기와 같이 구성되는 망관리 시스템에 사용되는 SNMP는 V1/ V2/ V3까지 사용되고 있는데, 그 중 SNMP V1/ V2의 접근 제한 방법으로는 "read-only"/"read-write" 커뮤니티(community)를 체크하는 방법이 주로 사용되고 있으며, SNMP V3는 보안 모듈이 프로토콜 내에 존재하고 있다.

<12> 커뮤니티는 매니저와 대리인간에 정의된 패스워드 방식의 내역을 의미한다.

<13> 예컨데, SNMP 버전1/ 버전2의 일반적인 커뮤니티는 "read-only"의 경우 "public", "read-write"의 경우 "private"로 많이 사용되고 있다. 또한, 어떤 시스템들은 이러한 커뮤니티들이 하드 코딩(hard-coding)되어 있어, 커뮤니티 변경이 용이하지 않다. 이와 같은 시스템들에 있어서는 커뮤니티 패스워드 자체가 노출되어 있어서 운용자 이외의 다른 사람이 임의의 장소에서 망 관리 시스템에 접근할 수 있게 되므로 보안상 문제가 발생할 수 있다.

【발명이 이루고자 하는 기술적 과제】

<14> 본 발명은, 상기와 같은 문제점을 해결하기 위하여 제안된 것으로, 현장에서 사용하고 있는 시스템 어플리케이션 프로토콜의 버전을 바꾸지 않고도 접근 제어(access control)를 할 수 있는 방법을 제공하는데 그 목적이 있다.

【발명의 구성 및 작용】

<15> 상기의 목적을 달성하기 위한 본 발명에 따른 망관리 시스템의 운용자 접근 제어에 대한 보안 방법은, 외부의 운용자가 TCP/IP(Transmission Control Protocol/Internet protocol) 또는 UDP/IP(User Datagram Protocol/Internet protocol)를 통해 IP(Internet protocol) 주소가

미리 설정되어 있는 IP 주소인지를 확인할 수 있도록 IP 필터링을 수행하는 제 1 단계; 및 상기 제 1 단계에 의해 통신 시스템에 접속시, 아이디(ID)/패스워드(Password) 입력 또는 커뮤니티(community)들을 설정하여 통신 시스템에 접속하는 제 2 단계를 포함하여 이루어지는 것을 특징으로 한다.

<16> 이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 일실시예를 상세히 설명한다.

<17> 본 발명에 적용되는 간이 망 관리 프로토콜(SNMP) 및 CLI(TL1)을 이용한 망관리 시스템은 종래기술과 동일하므로 그 설명은 생략한다.

<18> 도 3은 본 발명에 따른 망관리 시스템을 OSI 참조 모델과 관련하여 설명한 도면이다.

<19> 만약 TL1(110)을 이용하여 망관리 동작을 수행하는 경우, 도 1 및 도 3을 참조하면, 먼저 운용자는 사용자 인증을 위해 아이디(ID)와 패스워드(password)를 입력한다. 사용자 인증과정이 성공되면, TCP/IP 또는 UDP/IP를 통해 관리하고자 하는 시스템의 어플리케이션 계층에 접근한다. 이때 운용자가 사용하는 단말기의 IP 주소가 미리 설정된 IP 주소인지를 확인하기 위해, 보안 모듈(150)을 통해 어플리케이션 계층에 접근하도록 구성된다.

<20> 즉, IP 망(Ip network)(예컨데, 도 1 에서의 공중망)을 통한 원격 관리 채널(remote management channel)인 텔넷 터미널(400)은 아이디/패스워드 보안 장치에다 텔넷 프로토콜(telnet protocol)을 사용하는 운용 단말기(terminal)의 IP 주소(IP address)가 보안의 열쇠가 될 수 있는 필터링(filtering) 기능을 가진다.

<21> 여기서, 이 모듈은 "TL1" 기능을 구현한 "CLI(Command Line Interface)" 작업(task)과는 전혀 별개의 것으로 구현이 되어 있다.

<22> 그리고, SNMP V1/ V2는 기본적인 보안이 커뮤니티를 통해 이루어지는데, 이는 "read-only " 커뮤니티와 "read-write" 커뮤니티로 이루어지며 주로 변경이 가능하지 않는 경우가 많다.

<23> 본 발명에서는 이러한 커뮤니티의 보안을 위해, 각 커뮤니티는 "TL1" 명령어로만 변경이 가능하도록 한다. 즉, "SNMP"를 이용해서는 커뮤니티를 읽어볼 수도 없고, 변경도 블가능하므로 항상 운용자가 "TL1" 명령어를 통해 알고 있어야만 EMS 서버(500)와의 통신이 가능하다. 또한, 커뮤니티의 변경시에도 관리하는 EMS 서버(500)와의 협의가 요구되어진다.

<24> 그리고, SNMP V1/ V2가 UDP/IP나 TCP/IP를 사용할 경우 "TL1"과 마찬가지로 IP 필터링을 통해 운용자의 IP 주소를 키(key)로 하여 보안이 이루어지며 이것을 MIB으로 표현해 놓은 것이 표 1 내지 표 17이 된다.

<25> 표 1은 입력 패킷(ingress packets)을 필터링하기 위한 시스템의 정책 아이디를 표시한다. 이 객체의 값은 "entFilterPolicyTable"의 "entFilterPolicyId"의 값이다.

<26> 그리고, 'DEFVAL'는 모든 입력 패킷들을 인정한다.

<27> 【표 1】

```
entIngressFilterPolicyId OBJECT-TYPE
                SYNTAX      INTEGER (0..255)
                MAX-ACCESS  read-write
                STATUS      current
                DESCRIPTION
                    "
                        Indiactes the policy id of system for filtering ingress
packets.
                        The value of this object is that of entFilterPolicyId
                        in entFilterPolicyTable.

                        'DEFVAL' : accept all ingress packets
                    "
                DEFVAL { 0 }
                ::=  {entConfig 13}
```

<28> 그리고, 표 2는 출력 패킷(egress packets)을 필터링하기 위한 시스템의 정책 아이디를 표시한다. 이 객체의 값은 "entFilterPolicyTable"의 "entFilterPolicyId"의 값이다.

<29> 그리고, 'DEFVAL'는 모든 출력 패킷들을 버리지 않는다.

<30> 【표 2】

```
entEgressFilterPolicyId OBJECT-TYPE
                SYNTAX      INTEGER (0..255)
                MAX-ACCESS  read-write
                STATUS      current
                DESCRIPTION
                    "
                        Indiactes the policy id of system for filtering egress packets.
                        The value of this object is that of entFilterPolicyId
                        in entFilterPolicyTable.

                        'DEFVAL' : not discard all egress packets
                    "
                DEFVAL { 0 }
                ::= {entConfig 14 }
```

<31> 그리고, 표 3은 입/출력 패킷에 대한 시스템의 필터링 정책을 담고 있다. 이 표에서 "row"는 "entFilterIpTable"과 같은 프로토콜 표에서의 "row"를 가리킨다.

<32> 이 표에서 "row"를 생성하는데 있어서, "entFilterPolicyPointer" 객체에 의해 가리켜지는 "row"가 처음으로 생성된다.

<33> 그리고, 이 표에서 "row"를 삭제하는데 있어서, "entFilterPolicyPointer" 객체에 의해 가리켜지는 "row"가 처음으로 삭제된다.

<34>

【표 3】

```
entFilterPolicyTable OBJECT-TYPE
            SYNTAX  SEQUENCE OF EntFilterPolicyEntry
            MAX-ACCESS  not-accessible
            STATUS  current
            DESCRIPTION
            "
                 This table contains the filtering policies of system
                 on ingress/egress packet.
                 A row in this table is pointing a row in protocol table
                 such as entFilterIpTable.
                 For creating a row in this table, the row that is pointed
                 by entFilterPolicyPointer object was first created.
                 And for destroying a row in this table, the row that is pointed
                 by entFilterPolicyPointer object was first destroyed.
            "
            ::= { entConfig 15 }
```

<35> 그리고, 표 4에서 각각의 엔트리는 시스템의 필터링 정책을 대표하는 파라미터들의 리스트를 구성한다.

<36> 【표 4】

```
entFilterPolicyEntry  OBJECT-TYPE
            SYNTAX    EntFilterPolicyEntry
            MAX-ACCESS    not-accessible
            STATUS    current
            DESCRIPTION
            "
                 Each entry consists of a list of parameters that
                 represents filtering policy on a system.
            "
            INDEX     { entFilterPolicyIndex }
            ::= { entFilterPolicyTable 1 }

 EntFilterPolicyEntry ::= SEQUENCE {
             entFilterPolicyIndex        INTEGER,
             entFilterPolicyId           INTEGER,
             entFilterPolicyPointer      RowPointer,
             entFilterPolicyRowStatus    RowStatus
             }
```

<37> 그리고, 표 5는 "entFilterPolicyTable"로의 인덱스이다.

<38> 【표 5】

```
entFilterPolicyIndex OBJECT-TYPE
                SYNTAX      INTEGER(1..9)
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION
                    "
                        The index into the entFilterPolicyTable.
                    "
                ::= {entFilterPolicyEntry 1 }
```

<39> 그리고, 표 6은 입력 또는 출력 정책의 식별을 나타낸다. 같은 정책 아이디는 이 표에서 많은 "row"들에 속할 수 있다.

<40> 【표 6】

```
entFilterPolicyId OBJECT-TYPE
                SYNTAX      INTEGER(1..255)
                MAX-ACCESS  read-create
                STATUS      current
                DESCRIPTION
                    "
                        Indicates the identification of ingress or egress policy.
                        A same policy id could belong to many rows in this table.
                    "
                ::= { entFilterPolicyEntry 2 }
```

<41> 그리고, 표 7은 "entFilterIp"와 같은 프로토콜 표에서 "row"에 대한 포인터를 대표한다. 그 값은 프로토콜 표에서 첫번째 원주모양의 객체의 실예의 이름이다.

<42> 예를 들면, 이 객체의 실예의 값인 "entFilterIpIndex.3"은 "entFilterIp"표의 세번째 "row"를 가리킨다.

<43>

【표 7】

```
entFilterPolicyPointer OBJECT-TYPE
                SYNTAX      RowPointer
                MAX-ACCESS  read-create
                STATUS      current
                DESCRIPTION
                    "
                        Represents a pointer to a row in protocol table such as
 entFilterIp table. The value is the name of the instance of the first columnar object
in the protocol table.

                        For example, entFilterIpIndex.3 that is the value of the
instance of
                        this object would point to the 3rd row in the entFilterIp
table.
                    "
                ::=  {entFilterPolicyEntry 3 }
```

<44> 그리고, 표 8에서 객체는 새로운 "row"를 생성하거나, 수정하거나, 이 표에서 존재하는 "row"를 삭제하는데 이용된다.

<45> 만약, "entFilterIp"표와 같은 프로토콜 표의 관계된 "row"가 생성되지 않는다면, 이 표에서 "row"는 생성되어질 수 없을 것이다.

<46>

【표 8】

```
entFilterPolicyRowStatus OBJECT-TYPE
                SYNTAX      RowStatus
                MAX-ACCESS  read-create
                STATUS      current
                DESCRIPTION
                    "
                        This object is used to create a new row or modify or
                        delete an existing row in this table.

                        If the related row of protocol table such as entFilterIp table
wasn't created, a row in this table could have not been created.

                        The related row of protocol table should have been first
 destroyed before a row in this table is destroyed.
                    "
                ::= { entFilterPolicyEntry 4 }
```

<47> 그리고, 표 9는 IP 프로토콜 상의 필터 정책에 대한 세부 사항을 담고 있다.

<48> 【표 9】

```
entFilterIpTable OBJECT-TYPE
            SYNTAX  SEQUENCE OF EntFilterIpEntry
            MAX-ACCESS  not-accessible
            STATUS  current
            DESCRIPTION
            "
                This table contains the details of a filter policy over IP protocol.
            "
            ::= { entConfig 16 }
```

<49> 그리고, 표 10에서 각각의 엔트리는 IP 프로토콜 상의 필터 정책을 대표하는 파라미터들의 리스트를 구성한다.

<50>

【표 10】

```
entFilterIpEntry  OBJECT-TYPE
          SYNTAX    EntFilterIpEntry
          MAX-ACCESS    not-accessible
          STATUS    current
          DESCRIPTION
          "
               Each entry consists of a list of parameters that
               represents a filter policy over IP protocol .
          "
          INDEX    {  entFilterIpIndex }
          ::= { entFilterIpTable 1 }
EntFilterIpEntry ::= SEQUENCE {
           entFilterIpIndex            INTEGER,
           entFilterIp                 IpAddress,
           entFilterIpMask             IpAddress,
           entFilterIpPortNum          INTEGER,
           entFilterIpProtocol         INTEGER,
           entFilterIpControl          INTEGER,
           entFilterIpRowStatus        RowStatus
           }
```

<51> 그리고, 표 11은 "entFilterIpTable"로의 인덱스이다.

<52> 【표 11】

```
entFilterIpIndex OBJECT-TYPE
                SYNTAX      INTEGER(1..9)
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION
                    "
                        The index into the entFilterIpTable.
                    "
                ::= { entFilterIpEntry 1 }
```

<53> 그리고, 표 12는 필터 정책에 적용된 IP 주소를 지시한다.

<54>

【표 12】

```
entFilterIp OBJECT-TYPE
                SYNTAX      IpAddress
                MAX-ACCESS  read-create
                STATUS      current
                DESCRIPTION
                    "
                        Indicates ip address applied a filter policy.
                    "
                DEFVAL { '00000000'h }
                ::= { entFilterIpEntry 2 }
```

<55> 그리고, 표 13은 IP 주소의 마스크를 지시한다. 그리고, "entFilterIpProtocol"이 텔넷(telnet)일 때, 시스템은 항상 'DEFVAL'을 이 객체의 실시예에 적용한다.

<56> 【표 13】

```
entFilterIpMask OBJECT-TYPE
                SYNTAX      IpAddress
                MAX-ACCESS  read-create
                STATUS      current
                DESCRIPTION
                    "
                        Indicates the mask of ip address.
                        When entFilterIpProtocol is telnet,
                        system  always applies 'DEFVAL' to the instance of this object.
                    "
                DEFVAL { 'ffffffff'h }
                ::= { entFilterIpEntry 3 }
```

<57> 그리고, 표 14는 필터 정책에 적용되는 포트 넘버를 지시한다.

<58>

【표 14】

```
entFilterIpPortNum OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-create
                STATUS      current
                DESCRIPTION
                    "
                        Indicates the applied port number to a filter policy.
                    "
                ::= { entFilterIpEntry 4 }
```

<59> 그리고, 표 15는 필터 정책에 적용되는 프로토콜을 지시한다.

<60> 【표 15】

```
entFilterIpProtocol OBJECT-TYPE
                SYNTAX      INTEGER { snmp(1), telnet(2), tftp(3) }
                MAX-ACCESS  read-create
                STATUS      current
                DESCRIPTION
                    "
                        Indicates the applied protocol over IP protocol to a filter
policy.
                    "
                ::= { entFilterIpEntry 5 }
```

<61> 그리고, 표 16은 패킷을 받아들일지 버릴지를 결정한다.

<62> 【표 16】

```
entFilterIpControl OBJECT-TYPE
                SYNTAX      INTEGER { discard(1), accept(2) }
                MAX-ACCESS  read-create
                STATUS      current
                DESCRIPTION
                    "
                        Determines whether to discard or accept a packet.
                    "
                ::= { entFilterIpEntry 6 }
```

<63> 그리고, 표 17에서 이 객체는 새로운 "row"를 생성하거나, 수정하거나, 이 표에서 존재하는 "row"를 삭제하는데 이용된다.

<64> 【표 17】

```
entFilterIpRowStatus OBJECT-TYPE
                SYNTAX      RowStatus
                MAX-ACCESS  read-create
                STATUS      current
                DESCRIPTION
                    "
                        This object is used to create a new row or modify or
                        delete an existing row in this table.
                    "
                ::= { entFilterIpEntry 7 }
```

<65> 다음으로 SNMP V1/ V2를 위한 커뮤니티 변경 및 조회에 관한 "TL1" 명령어의 수행 결과는 표 18 과 같다.

<66> 【표 18】

```
SU-WON>   rtrv-community;

IP C01240
<


   SU-WON 2002-02-02 01:56:40
M  C01240 COMPLD
   "RD=SamsungAcemap,WR=K_SAMSUNG_Acemap2000_set,TR=SS_Acemap_Trap"
   /* RTRV-COMMUNITY; [C01240] */
;
```

<67> 여기서, "RD"는 "read-only" 커뮤니티, "WR"은 "read-write" 커뮤니티, "TR"은 "트랩(trap)" 커뮤니티를 의미한다. 이것은 "TL1" 명령어로만 변경 및 조회가 가능하고, 변경시 EMS 서버(500)에서의 관리를 위해서는 EMS 서버(500)에서도 커뮤니티의 변경이 되어야만 한다.

<68> 다음은 필터링에 관한 동작을 표 1 내지 표 17에 표현되어 있는 MIB 객체로 설명하면 다음과 같다. 먼저 "entFilterIpTable"에 있는 객체들을 필터링시키고자하는 범위로 셋팅(setting)한 후 "row"를 생성(create)시킨다. 이때, "entFilterIpProtocol"의 의미는 "IP 상의 프로토콜(protocol over ip)"라고도 표현되어질 수 있다.

<69> 여기서, 필터링하고자 하는 프로토콜은 SNMP/ 텔넷/ TFTP(Trivial File Transfer Protocol) 등이 가능하다. 그리고, "entFilterIpControl"에서는 패킷을 버리거나(discard) 받아들이는(accept) 것이 설정가능한 값이다.

<70> 해당 "row"가 출력 정책(egress policy)으로 사용되어질 경우에, SNMP 패킷은 요청(request)은 받아들여지나 응답 패킷(response packet)이 내보내지지 않는다. 물론 트랩에도 마찬가지로 적용되어져 트랩 패킷도 등록된 EMS 서버(500)로 내보내지지 않게 된다. 반면, 해당 "row"가 입력 정책(ingress policy)으로 사용되어질 경우에는 반대로 동작하게 된다. 일단 "entFilterIpTable"의 "row"가 생성되고 난 후에는 "entFilterPolicyTable"의 "row"가 생성되어야 한다. 이 표(table)의 목적은 하나의 정책에 여러 가지 "row"들이 포함될 수도 있는 다양성을 위해 구현된다.

<71> 그리고, "entFilterPolicyPointer"는 앞서 만들어진 "entFilterIpTable"의 "row"를 포인팅(pointing)하고 있다. 이 때 "entFilterPolicyId"는 같은 값을 여러개의 "row"들이 가지는 것도 가능한 구조로 구현된다. 그리고, "entIngressFilterPolicyId"와 "entEgressFilterPolicyId"의 값을 셋팅한다. 이 값들은 시스템과 타 장비간의 주고받는 패킷 전반에 영향을 미치게 된다.

<72> 이상 본 발명의 바람직한 실시예에 대해 상세히 기술되었지만, 본 발명이 속하는 기술분야에 있어서 통상의 지식을 가진 사람이라면, 첨부된 청구 범위에 정의된 본 발명의 정신 및 범위를 벗어나지 않으면서 본 발명을 여러 가지로 변형 또는 변경하여 실시할 수 있음을 알 수 있을 것이다. 따라서 본 발명의 앞으로의 실시예들의 변경은 본 발명의 기술을 벗어날 수 없을 것이다.

【발명의 효과】

<73> 이상 설명한 바와 같이, 본 발명에 따르면, 망관리 프로토콜 버전이 EMS와 동일 버전으로 운용 되고 있는 시스템에서, SNMP V1/ V2을 보안 기능이 제공되는 SNMP V3로 업그레이드하지 않고도 IP 필터링을 수행하는 보안 모듈을 추가함으로써 간단하게 망관리 인터페이스에 접속시 보안을 유지할 수 있게 된다.

【특허청구범위】

【청구항 1】

외부의 운용자가 TCP/IP(Transmission Control Protocol/Internet protocol) 또는 UDP/IP(User Datagram Protocol/Internet protocol)를 통해 IP(Internet protocol) 주소가 미리 설정되어 있는 IP 주소인지를 확인할 수 있도록 IP 필터링을 수행하는 제 1 단계; 및

상기 제 1 단계에 의해 통신 시스템에 접속시, 아이디(ID)/패스워드(Password) 입력 또는 커뮤니티(community)들을 설정하여 통신 시스템에 접속하는 제 2 단계를 포함하여 이루어지는 망관리 시스템의 운용자 접근 제어에 대한 보안 방법.

【청구항 2】

제 1 항에 있어서, 상기 제 1 단계는,

아래 표 19 및 표 20의 MIB(Management Information Base)으로 구현된 객체들을 필터링 시키고자하는 범위로 셋팅(setting)한 후 "row"를 생성(create)시키는 제 11 단계;

입력되거나 출력하려는 SNMP 패킷에 대해 버리거나(discard) 받아들이는(accept) 것이 설정가능하도록 하는 제 12 단계;

상기 "row"가 출력 정책(egress policy)으로 사용되어질 경우에, 상기 SNMP 패킷은 요청(request)은 받아들여지나 응답 패킷(response packet)은 출력하지 않도록 하는 제 13 단계; 및

상기 "row"가 입력 정책(ingress policy)으로 사용되어질 경우에, 상기 SNMP 패킷은 요청(request)은 받아들여지지 않으나 응답 패킷(response packet)은 출력하도록 하는 제 14 단계를 포함하는 망관리 시스템의 운용자 접근 제어에 대한 보안 방법.

【표 19】

```
entFilterIpTable OBJECT-TYPE
            SYNTAX  SEQUENCE OF EntFilterIpEntry
            MAX-ACCESS  not-accessible
            STATUS  current
            DESCRIPTION
            "
                This table contains the details of a filter policy over IP protocol.
            "
            ::= { entConfig 16 }
```

【표 20】

```
entFilterIpEntry  OBJECT-TYPE
            SYNTAX    EntFilterIpEntry
            MAX-ACCESS    not-accessible
            STATUS    current
            DESCRIPTION
            "
                Each entry consists of a list of parameters that
                represents a filter policy over IP protocol .
            "
            INDEX    {  entFilterIpIndex }
            ::= { entFilterIpTable 1 }
EntFilterIpEntry ::= SEQUENCE {
             entFilterIpIndex            INTEGER,
             entFilterIp                 IpAddress,
             entFilterIpMask             IpAddress,
             entFilterIpPortNum          INTEGER,
             entFilterIpProtocol         INTEGER,
             entFilterIpControl          INTEGER,
             entFilterIpRowStatus        RowStatus
             }
```

【청구항 3】

제 1 항 또는 제 2 항에 있어서, 상기 외부의 운용자는 텔넷 터미널 또는 EMS 서버인 것을 특징으로 하는 망관리 시스템의 운용자 접근 제어에 대한 보안 방법.

【청구항 4】

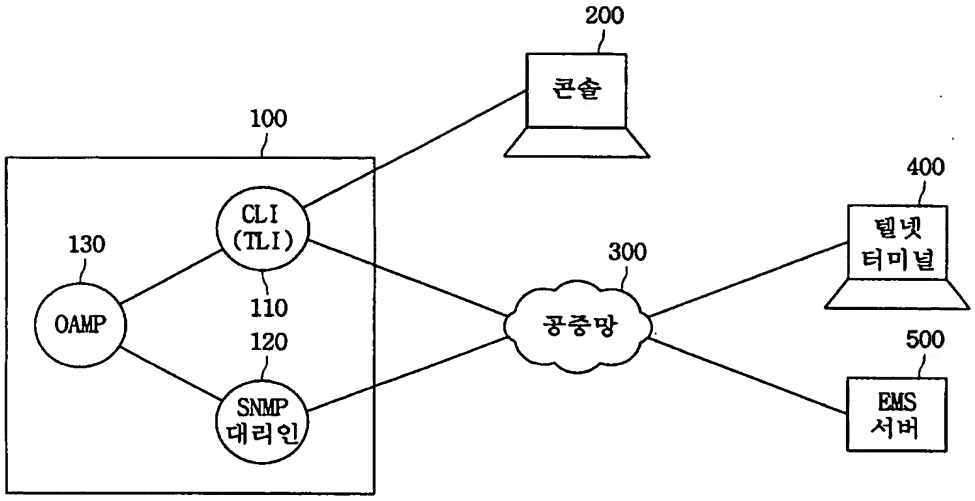
제 2 항에 있어서, 상기 제 12 단계의 SNMP 패킷을 버리거나(discard) 받아들이는(accept) 것의 설정이 가능하도록 아래 표 21에서 MIB으로 구현된 것과 같이 정의하는 것을 특징으로 하는 망관리 시스템의 운용자 접근 제어에 대한 보안 방법.

【표 21】

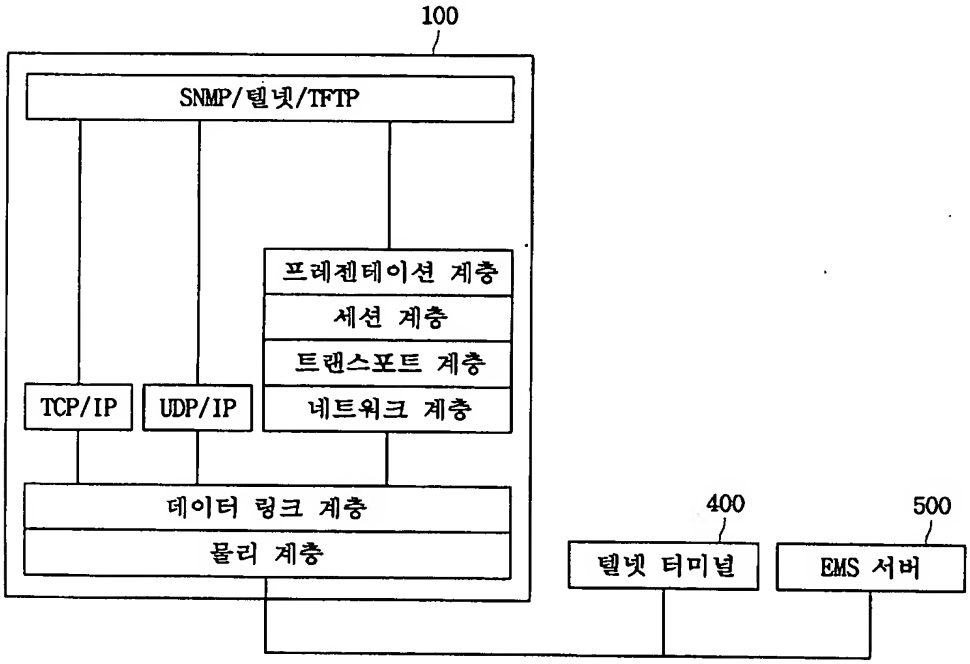
```
entFilterIpControl OBJECT-TYPE
                SYNTAX      INTEGER { discard(1), accept(2) }
                MAX-ACCESS  read-create
                STATUS      current
                DESCRIPTION
                    "
                        Determines whether to discard or accept a packet.
                    "
                ::= { entFilterIpEntry 6 }
```

【도면】

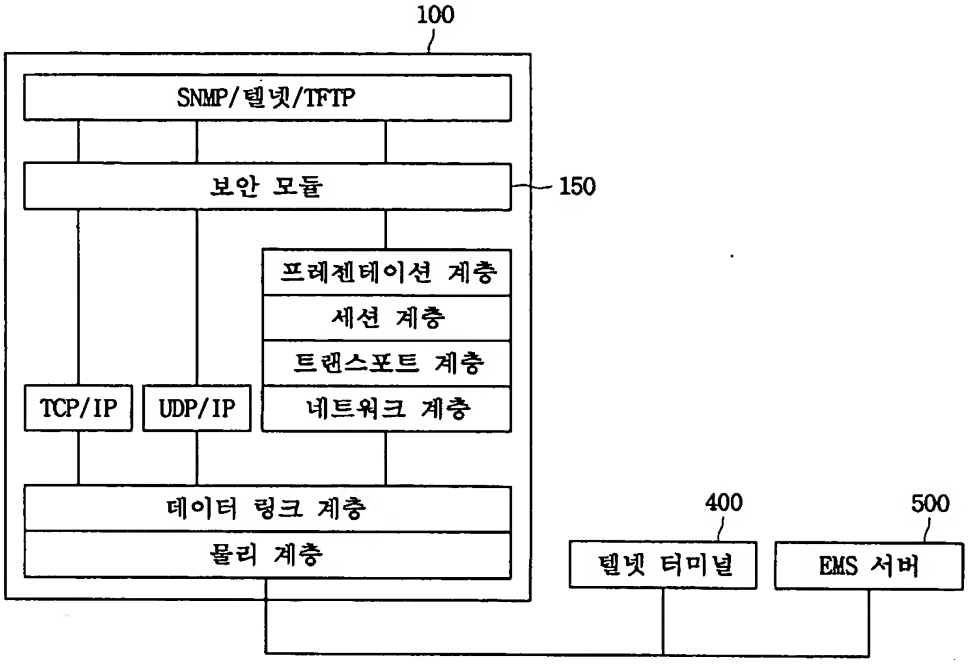
【도 1】


200
콘솔
100
CLI
(TLI)
130
OAMP
110
120
SNMP
대리인
300
공중망
400
텔넷
터미널
500
EMS
서버

【도 2】


100
SNMP/텔넷/TFTP
프레젠테이션 계층
세션 계층
트랜스포트 계층
네트워크 계층
TCP/IP
UDP/IP
데이터 링크 계층
물리 계층
400
텔넷 터미널
500
EMS 서버

【도 3】


100
SNMP/텔넷/TFTP
보안 모듈
150
프레젠테이션 계층
세션 계층
트랜스포트 계층
TCP/IP
UDP/IP
네트워크 계층
데이터 링크 계층
물리 계층
400
텔넷 터미널
500
EMS 서버